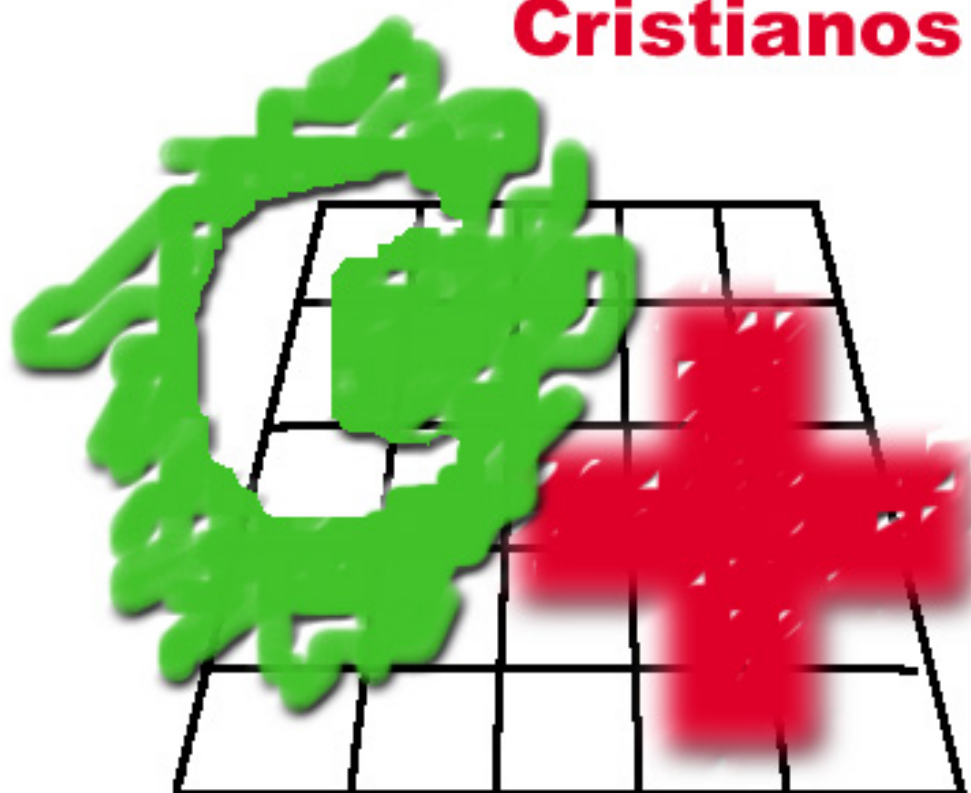


LÓGICA DE PRIMER ORDEN

MOROS

y

Cristianos



© 1998 Ver. 1.11

Departamento de *Ciencia de la Computación
e Inteligencia Artificial*
Universidad de Alicante

Herramienta didáctica para el aprendizaje práctico de la Lógica de Primer Orden:

Moros y Cristianos

Faraón Llorens Largo
Diciembre 1998

DEPARTAMENTO DE *CIENCIA DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL*
Universidad de Alicante



Faraón Llorens Largo
Departamento de *Ciencia de la Computación e Inteligencia Artificial*
Universidad de Alicante
<http://www.dccia.ua.es/logica>
logica@dccia.ua.es

Diciembre 1998

Palabras clave: Lógica de Primer Orden, Fórmula Bien Formada, Valoración de Verdad.

Resumen

Este informe presenta la herramienta didáctica para el aprendizaje de la lógica de primer orden *Moros y Cristianos (MyC)*. Este programa ha sido desarrollado en el departamento de *Ciencia de la Computación e Inteligencia Artificial* de la Universidad de Alicante, bajo la dirección de los profesores Faraón Llorens Largo y M^a Jesús Castel de Haro, y programado por Eugenio Arjones Cano y Alfredo Soro Busto. Se trata de un programa de apoyo dirigido a los alumnos de la asignatura *Lógica de Primer Orden* [Llorens96], [Llorens98], orientado al análisis de fórmulas lógicas, tanto en el aspecto sintáctico como en el semántico, en un intento de consolidar los conceptos teóricos y asentarlos mediante un entorno amigable.

En los últimos años ha habido un creciente interés por la lógica desde el campo de la informática, ya que esta aporta un método formal para la gestión del conocimiento (representación y razonamiento). Al mismo tiempo, los informáticos han desarrollado distintos programas que ayudan a los estudiantes a adquirir estas técnicas formales [Goldson93]. Los ordenadores nos proporcionan una excelente herramienta para entender, de forma práctica, algunas de las nociones fundamentales de la lógica así como la forma de representar y razonar [Barwise98]. El programa *MyC* está basado en el programa *Tarski's World* [Barwise92], entorno desarrollado en la universidad de Stanford que sirve de soporte para una nueva aproximación pedagógica a la enseñanza de la lógica, permitiendo construir mundos tridimensionales de bloques y describirlos en el lenguaje simbólico de la lógica de primer orden.

El programa permite la introducción de fórmulas lógicas y su análisis sintáctico para determinar si son *fórmulas bien formadas (fbf)*. Estas fórmulas pueden ser evaluadas semánticamente en *mundos* (interpretaciones) formados por individuos que pueden ser *moros* o *cristianos*, y que a su vez pueden tener algunas propiedades (*llevar armadura, tener espada y montar a caballo*). Esto permite al alumno que se inicia en el aprendizaje de la lógica entender de forma práctica los conceptos lógicos, conceptos que por sí solos son de naturaleza eminentemente abstracta y por ello difíciles de entender.

El Contenido

<i>El Contenido</i>	<i>I</i>
<i>El Entorno</i>	<i>I</i>
1. Introducción.....	1
2. Instalación y Requerimientos	1
3. Manejo del Programa	2
<i>El Lenguaje</i>	3
1. Descripción.....	3
2. Ejercicios	5
1. Evaluación Sintáctica de fórmulas sin cuantificación	5
2. Evaluación Sintáctica de fórmulas con cuantificación	6
3. Traducción de fórmulas a lenguaje natural.....	6
<i>Los Mundos</i>	6
1. Descripción.....	6
2. Ejercicios	9
1. Evaluación Semántica de fórmulas sin cuantificación	9
2. Evaluación Semántica de fórmulas con cuantificación	10
3. Clasificación semántica de fórmulas	11
4. Análisis semántico de fórmulas.....	12
<i>La Práctica</i>	12
1. Equivalencia de fórmulas	12
<i>La Tabla de las Figuras</i>	14
<i>La Bibliografía</i>	14

El Entorno

1. Introducción

Si algo caracteriza este final de siglo ¡y de milenio! ha sido sin duda la introducción en nuestras vidas de los ordenadores. Tenemos ordenadores que nos ayudan a hacer “casi todo”. En esta línea argumental, hemos desarrollado una herramienta que favorezca el aprendizaje de la lógica, de forma que “practicando” lógica sea como aprendamos lógica.

En los últimos años ha habido un creciente interés por la lógica desde el campo de la informática, ya que esta aporta un método formal para la gestión del conocimiento (tanto en el aspecto de representación como de razonamiento). Como contrapartida, los informáticos han desarrollado distintos programas que ayudan a los estudiantes a adquirir estas técnicas formales [Goldson93]. Los ordenadores nos proporcionan una excelente herramienta para entender, de forma práctica, algunas de las nociones fundamentales de la lógica así como la forma de representar y razonar [Barwise98]. El programa *Moros y Cristianos* (MyC) está basado en el programa *Tarski's World* [Barwise92], entorno desarrollado en la universidad de Stanford que sirve de soporte para una nueva aproximación pedagógica a la enseñanza de la lógica, permitiendo construir mundos tridimensionales de bloques y describirlos en el lenguaje simbólico de la lógica de primer orden.

MyC permite introducir fórmulas lógicas y evaluarlas sintácticamente determinando si son *fbf* (fórmulas bien formadas). Al mismo tiempo se pueden crear *mundos* (interpretaciones) que nos permitirán evaluar semánticamente dichas fbf. La ventaja estriba en que estos mundos ya no son abstractos, sino que son tangibles y reales, y estarán definidos por el alumno que podrá añadir, quitar y modificar sus componentes. Estos individuos de nuestros mundos podrán ser *moros* o *cristianos*, que a su vez pueden tener distintas características, como por ejemplo *llevar armadura*, *tener espada* y *montar a caballo*.

2. Instalación y Requerimientos

Moros y Cristianos (MyC) ha sido desarrollado en el departamento de *Ciencia de la Computación e Inteligencia Artificial* de la Universidad de Alicante, bajo la dirección de los profesores Faraón Llorens Largo y M^a Jesús Castel de Haro, y programado por Eugenio Arjones Cano y Alfredo Soro Busto. Se trata de un programa didáctico de apoyo a la docencia, dirigido fundamentalmente a los alumnos de las asignaturas *Lógica de Primer Orden* de las distintas titulaciones en las que se imparte (Ingeniero Técnico en Informática de Gestión, Ingeniero Técnico en Informática de Sistemas, Ingeniero en Informática y

Licenciado en Matemáticas). El programa está orientado al análisis de fórmulas lógicas, tanto en el aspecto sintáctico como en el semántico, en un intento de consolidar los conceptos teóricos y asentarlos mediante un entorno amigable.

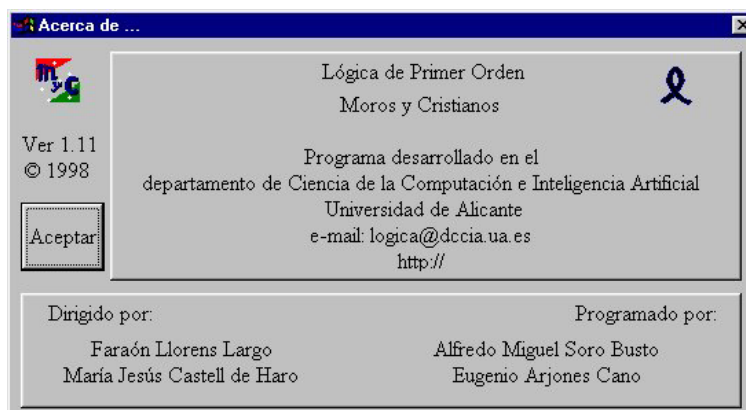


Ilustración 1: Ventana "Acerca de ..."

Los requerimientos técnicos mínimos para que el programa pueda funcionar son:

- Windows 95
- CPU 486 a 66 Mhz o superior
- 8 Mb de memoria RAM
- 5 Mb de espacio libre en el Disco Duro
- Tarjeta de sonido compatible (opcional)

El programa se encuentra disponible en la dirección

<http://www.dccia.ua.es/logica/MyC>

y se puede utilizar libremente con fines educativos. Para cualquier sugerencia sobre el mismo os podéis dirigir a la dirección de correo electrónico

logica@dccia.ua.es

3. Manejo del Programa

El programa permite la introducción de fórmulas lógicas y su análisis sintáctico para determinar si son *fórmulas bien formadas (fbf)*. Estas fórmulas pueden ser evaluadas semánticamente en *mundos* formados por individuos que pueden ser *moros* o *cristianos*, y que a su vez pueden tener algunas propiedades (*llevar armadura*, *tener espada* y *montar a caballo*). Esto permite al alumno que se inicia en el aprendizaje de la lógica entender de forma práctica los conceptos lógicos, conceptos que por sí solos son de naturaleza eminentemente abstracta y por ello difíciles de entender.

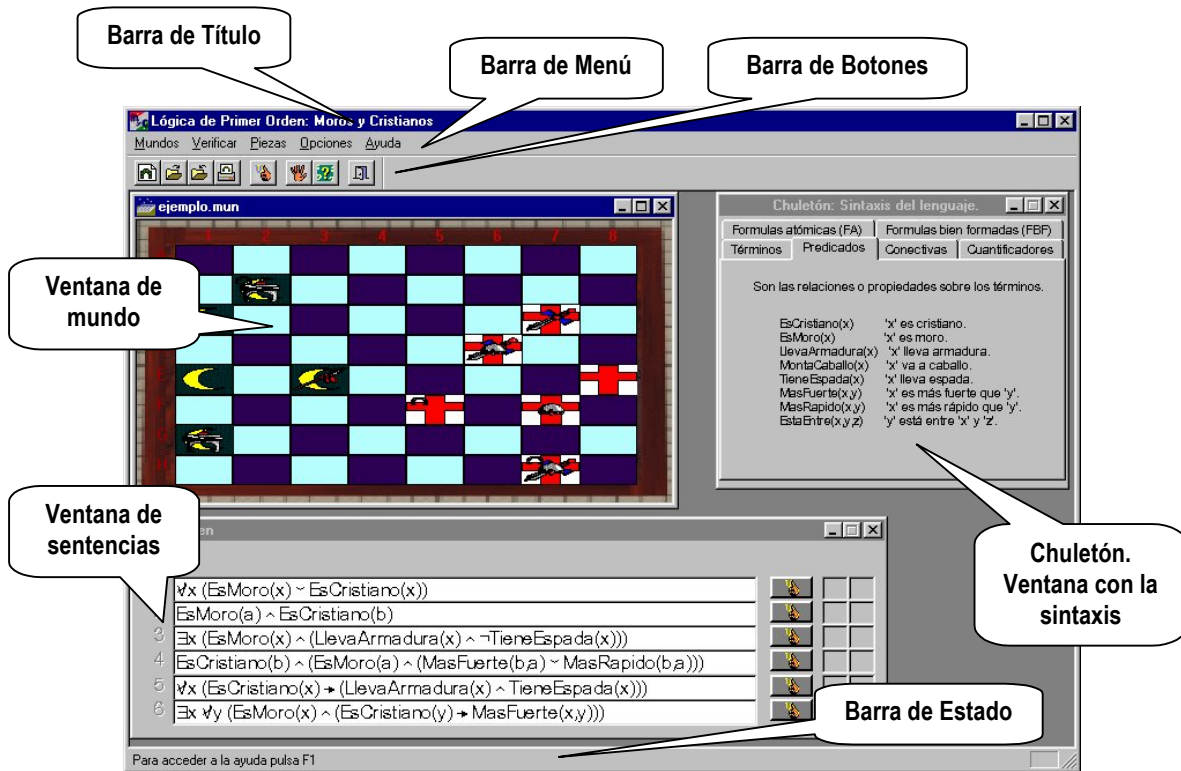


Ilustración 2: Pantalla principal del programa

El Lenguaje

1. Descripción

En un primer momento vamos a trabajar con el *aspecto sintáctico* de la lógica. El lenguaje que vamos a utilizar para representar nuestras sentencias (nuestro conocimiento y visión del mundo en el que trabajaremos) será el *lenguaje de la lógica de primer orden* [Llorens96], [Llorens98]. Adaptaremos y restringiremos el alfabeto a las características de los posibles mundos que podemos representar (formados por moros y cristianos con sus respectivas propiedades particulares). El *alfabeto* a utilizar quedará, por tanto, formado por:

- Símbolos de *Términos*: representan a los objetos de nuestro mundo, que tomarán valores en el **Universo del Discurso**. Estos términos pueden ser
 - ✓ **Constantes** : son objetos concretos del universo del discurso. Utilizaremos las letras:

a, a0, a1, ..., a9, b, b0, b1, ..., b9, c, c0, c1, ..., c9

- ✓ **Variables** : son objetos cualesquiera del universo del discurso. Utilizaremos las letras:

$x, x_0, x_1, \dots, x_9, y, y_0, y_1, \dots, y_9, z, z_0, z_1, \dots, z_9$

- Símbolos de *Predicados*: representan las relaciones o propiedades que se pueden dar en nuestro mundo y que se aplican sobre los términos. En concreto tenemos:

- ✓ **EsCristiano**(x) x es cristiano
- ✓ **EsMoro**(x) x es moro
- ✓ **LlevaArmadura**(x) x lleva una armadura
- ✓ **MontaCaballo**(x) x va montado a caballo
- ✓ **TieneEspada**(x) x va armado con una espada
- ✓ **MasFuerte**(x, y) x es más fuerte que y
- ✓ **MasRapido**(x, y) x es más rápido que y
- ✓ **EstaEntre**(x, y, z) y está situado entre x y z

- *Conectivas*: son símbolos lógicos que conectan fórmulas, formando formulas más complejas. Disponemos de las siguientes conectivas lógicas:

- ✓ **Negación** $\neg$ (Ctrl+n)
- ✓ **Conjunción** $\wedge$ (Ctrl+y)
- ✓ **Disyunción** $\vee$ (Ctrl+o)
- ✓ **Implicación** $\rightarrow$ (Ctrl+i)

- *Cuantificadores*: son símbolos lógicos que nos permiten expresar la cantidad de objetos que satisfacen cierta condición. Los cuantificadores lógicos de que disponemos son:

- ✓ **Universal** $\forall$ (Ctrl+t)
- ✓ **Existencial** $\exists$ (Ctrl+e)

Combinando adecuadamente los símbolos de nuestro alfabeto podemos formar las palabras de nuestro lenguaje. A las palabras sintácticamente correctas las llamaremos *fórmulas atómicas*. Así, definimos:

- *Fórmula Atómica*: son fórmulas lógicas formadas por un símbolo de predicado seguido de n argumentos ocupados por los correspondientes términos. Por ejemplo, son fórmulas atómicas:

- | | |
|--------------------------------|--------------------------------------|
| EsCristiano (x) | “ x es cristiano” |
| EsMoro (a) | “el individuo a es un moro” |
| LlevaArmadura (b) | “ b lleva una armadura” |
| MontaCaballo (y) | “ y va montado a caballo” |
| TieneEspada (c) | “ c va armado con una espada” |
| MasFuerte (a, b) | “ a es más fuerte que b ” |
| MasRapido (x, a) | “ x es más rápido que a ” |
| EstaEntre (x, a, z) | “ a está situado entre x y z ” |

Combinando las palabras de forma adecuada podemos construir las frases de nuestro lenguaje. A las frases sintácticamente correctas las llamaremos *fórmulas bien formadas* (*fbf*). Las reglas para la construcción de fbf son:

- *Fórmula Bien Formada (fbf)*: son fórmulas lógicas construidas según las siguientes reglas:
 1. Toda fórmula atómica es fbf
 2. Si F_1 y F_2 son fbf entonces
 - (F_1) es una fbf
 - $\neg F_1$ es una fbf
 - $F_1 \wedge F_2$ es una fbf
 - $F_1 \vee F_2$ es una fbf
 - $F_1 \rightarrow F_2$ es una fbf
 3. Si F es una fbf con la variable x libre entonces
 - $\forall x F$ es una fbf
 - $\exists x F$ es una fbf
 4. Únicamente son fbf las construidas con los pasos 1 a 3

Para evaluar sintácticamente las fórmulas (determinar si son *fbf*) disponemos de la opción de “Verificar” fórmulas o simplemente pulsando el botón que hay al lado de la sentencia.

2. Ejercicios

Veamos algunos ejercicios que podemos resolver ayudados por el programa MyC.

1. Evaluación Sintáctica de fórmulas sin cuantificación

Escribe las *fórmulas bien formadas* correspondientes a las siguientes sentencias:

- El individuo a es moro o es cristiano
- a es un moro y b es un cristiano
- a es un moro a caballo con armadura pero que no lleva espada
- El cristiano b es más fuerte o más rápido que el moro a
- Si b es cristiano entonces lleva armadura y espada
- a es moro y además si b es cristiano entonces es más fuerte a que b .

Evalúalas sintácticamente para comprobar que realmente están bien escritas y las guardas en el fichero “practica1.sen”.

2. Evaluación Sintáctica de fórmulas con cuantificación

Escribe las *fórmulas bien formadas* correspondientes a las siguientes sentencias:

- Todos son moros o todos son cristianos
- Todos son moros o cristianos
- Al menos existe un moro y un cristiano
- Todos los cristianos sin armadura llevan espada
- Para cada cristiano hay un moro que es más fuerte o rápido que él
- Algún moro es más fuerte que todos los cristianos

Evalúalas sintácticamente para comprobar que realmente están bien escritas y las guardas en el fichero “practica2.sen”.

3. Traducción de fórmulas a lenguaje natural

Escribe las siguientes *fórmulas bien formadas* en la ventana de sentencias y las guardas en el fichero “practica3.sen”. Escribe una frase en lenguaje natural (castellano) para cada una de ellas :

$$\exists x (EsMoro(x) \wedge \forall y (EsCristiano(y) \rightarrow MasFuerte(x,y)))$$

$$\exists x \forall y (EsMoro(x) \wedge (EsCristiano(y) \rightarrow MasFuerte(x,y)))$$

$$\exists x \forall y ((EsMoro(x) \wedge EsCristiano(y)) \rightarrow MasFuerte(x,y))$$

Los Mundos

1. Descripción

Los mundos nos permitirán trabajar con el *aspecto semántico* de la lógica. Para poder evaluar semánticamente las fórmulas bien formadas que hemos escrito antes (estudiar el significado y el valor de verdad de dichas fbf), necesitamos un mundo donde interpretarlas. Para ello, en lugar de trabajar en abstracto, con cualquier mundo pensable (cualquier dominio o universo del discurso), adoptaremos un enfoque minimalista de forma que “cerraremos” el mundo donde evaluamos nuestras sentencias. Por tanto asumimos que los únicos individuos que existen y las únicas propiedades que se cumplen son las que aparecen explícitamente representadas en el mundo.

Nuestro *mundo* se limita a un tablero de 8 x 8 posiciones (64 casillas) donde podemos situar a nuestros individuos. Cada posición vendrá determinada por la fila (de la A a la H) y la columna (del 1 al 8).



Ilustración 3: Ventana de mundo: Tablero

Estos individuos que podremos incluir en nuestro mundo podrán ser exclusivamente *moros* o *cristianos* y podrán tener alguna de las características permitidas (*espada*, *armadura* y *caballo*). La representación gráfica para los distintos tipos de individuos de que disponemos la podemos ver en la siguiente figura.

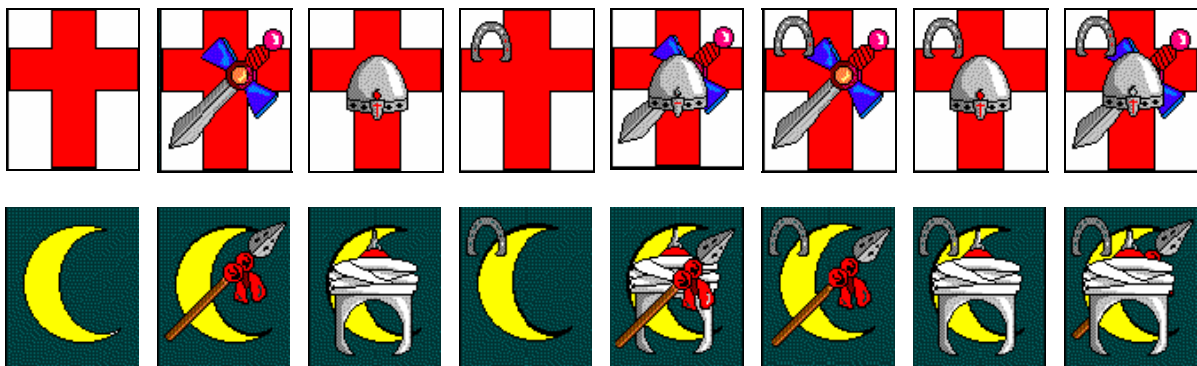


Ilustración 4: Tipos de individuos

En el tablero podremos insertar un individuo mediante la opción del menú “Insertar pieza”. A dicho objeto le tenemos que dar obligatoriamente un nombre que lo identifique de manera inequívoca (términos constantes definidos en el lenguaje):

a, a0, a1, ..., a9, b, b0, b1, ..., b9, c, c0, c1, ..., c9



Ilustración 5: Insertar una nueva pieza



Ilustración 6: Tablero con pieza

También podemos insertar una nueva pieza haciendo click con el botón derecho del ratón sobre la casilla deseada, con lo que nos evitamos tener que introducir las coordenadas de la posición en el tablero.

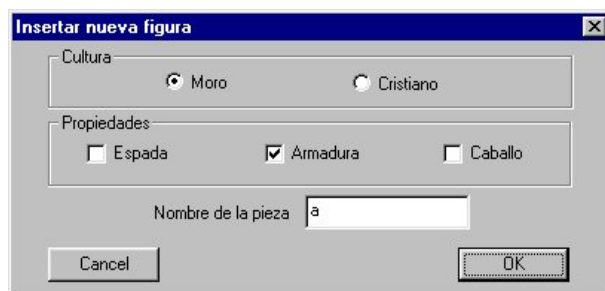


Ilustración 7: Insertar pieza desde tablero

Para modificar las características de cualquier pieza basta con pulsar el botón derecho del ratón sobre la misma, con lo que abrirá la ventana de insertar nueva pieza con

las características que tiene en este momento. Modificando esta información actualizaremos a los valores nuevos que deseemos.

Podemos cambiar las piezas de posición si pulsamos el botón izquierdo del ratón sobre la pieza y, sin soltarlo, la desplazamos hasta la nueva casilla. Para eliminar de nuestro mundo un determinado individuo disponemos de la opción del menú “Quitar pieza” que nos pedirá la posición de la pieza que queremos eliminar.

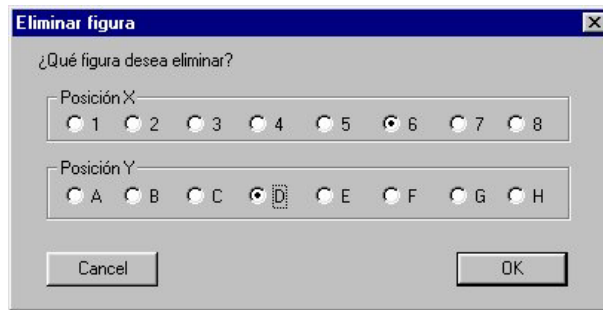


Ilustración 8: Eliminar pieza

2. Ejercicios

Veamos algunos ejercicios que nos permitirán evaluar semánticamente las fórmulas.

1. Evaluación Semántica de fórmulas sin cuantificación

Abre un mundo nuevo y busca un *modelo* (interpretación que hace verdadera la fórmula) y un *contramodelo* (interpretación que hace falsa la fórmula) para cada una de las sentencias del ejercicio 1 del apartado anterior (practical.sen). Para ello inserta y/o elimina los individuos que creas necesarios.

¿Puedes encontrar un mundo donde todas sean ciertas al mismo tiempo? Si la respuesta anterior es negativa, ¿qué sentencias son las que entran en contradicción?

Cambia alguna sentencia para hacer que en dicho mundo todas las sentencias sean verdaderas al mismo tiempo (que sea modelo del conjunto completo de sentencias) y lo guardas en el fichero “practical.mun”.

Como ejemplo, en la siguiente tabla aparece un mundo donde todas las sentencias del ejercicio 1 (con la sentencia 6 modificada “MasFuerte(a,b)” por “MasFuerte(b,a)”) son ciertas.

Nombre	Cultura	Espada	Armadura	Caballo	
<i>a</i>	moro		✓	✓	
<i>b</i>	cristiano	✓	✓		

La sentencia 5 “Si *b* es cristiano entonces lleva armadura y espada” es cierta ya que en nuestro mundo hemos incluido un cristiano llamado *b* que lleva armadura y espada (para que se cumplan las sentencias 2 y 5); por otro lado nuestro mundo también incluye un individuo *a* que es moro y no tiene espada (para que sean ciertas las sentencias 2 y 3). En este mismo mundo ¿qué valor tendría la sentencia “Si *a* es cristiano entonces lleva armadura y espada”? (razona la respuesta)

*Sería verdadera, ya que la implicación también es cierta si el antecedente es falso, y como *a* no es cristiano (es moro), toda la sentencia sería verdadera.*

2. Evaluación Semántica de fórmulas con cuantificación

Abre un mundo nuevo y busca un *modelo* (interpretación que hace verdadera la fórmula) y un *contramodelo* (interpretación que hace falsa la fórmula) para cada una de las sentencias del ejercicio 2 del apartado anterior (practica2.sen). Para ello inserta y/o elimina los individuos que creas necesarios.

¿Puedes encontrar un mundo donde todas sean ciertas al mismo tiempo? Si la respuesta anterior es negativa, ¿qué sentencias son las que entran en contradicción?

Cambia alguna sentencia para hacer que en dicho mundo todas las sentencias sean verdaderas al mismo tiempo (que sea modelo del conjunto completo de sentencias) y guardalo en el fichero “practica2.mun”.

Por ejemplo, en la siguiente tabla aparece un mundo donde todas las sentencias del ejercicio 2 (con la sentencia 1 modificada cambiando la conjunción por el implicador $\forall x \text{ EsMoro}(x) \rightarrow \forall y \text{ EsCristiano}(y)$) son ciertas.

Nombre	Cultura	Espada	Armadura	Caballo	
<i>a</i>	moro	✓	✓	✓	
<i>b</i>	cristiano		✓		

3. Clasificación semántica de fórmulas

¿Puedes encontrar un contramodelo de la primera sentencia del ejercicio 1 "El individuo *a* es moro o cristiano"? ¿Por qué? ¿Cómo calificarías a dicha fórmula?

*No, no se puede encontrar un contramodelo de la sentencia "el individuo *a* es moro o es cristiano" ya que en los mundos que podemos crear con este programa cada individuo que añadimos a nuestro mundo debe ser obligatoriamente o moro o cristiano. Una posibilidad sería que en nuestro mundo no hubiese ningún individuo con nombre *a*, pero en este caso al evaluarla semánticamente da error y nos dice que no puede evaluar semánticamente la fórmula ya que la constante *a* no está definida en el mundo. La fórmula será, por tanto, una TAUTOLOGÍA.*

Escribe una fórmula sin cuantificación que sea CONTRADICCIÓN, una que sea TAUTOLOGÍA y una que sea INDETERMINACIÓN en los mundos posibles que se pueden crear con este programa.

Las más sencillas serían:

<i>Contradicción</i>	$EsMoro(a) \wedge EsCristiano(a)$
<i>Tautología</i>	$EsMoro(a) \vee EsCristiano(a)$
<i>Indeterminación</i>	$EsMoro(a)$

¿Puedes encontrar un contramodelo de la segunda sentencia del ejercicio 2 "Todos son moros o cristianos"? ¿Por qué? ¿Cómo calificarías a dicha fórmula?

No, no se puede encontrar un contramodelo de la sentencia "Todos son moros o cristianos" ya que, al igual que antes, en los mundos que podemos crear con este programa cada individuo que añadimos a nuestro mundo debe ser obligatoriamente o moro o cristiano. La fórmula será, por tanto, una TAUTOLOGÍA.

Escribe una fórmula cuantificada que sea CONTRADICCIÓN, una que sea TAUTOLOGÍA y una que sea INDETERMINACIÓN en los mundos posibles que se pueden crear con este programa.

Las más sencillas serían:

<i>Contradicción</i>	$\forall x \text{ EsMoro}(x) \wedge \exists y \text{ EsCristiano}(y)$
<i>Tautología</i>	$\exists x (\text{EsMoro}(x) \vee \text{EsCristiano}(x))$
<i>Indeterminación</i>	$\exists x \text{ EsMoro}(x)$

4. Análisis semántico de fórmulas

Abre el fichero “practica3.sen” (sentencias introducidas en el ejercicio 3 del apartado anterior). Analiza las tres fórmulas, estudiando la semejanza o diferencia entre ellas. Para ello te puede ayudar crear mundos donde evaluarlas. Crea un mundo donde no haya moros, ¿qué ocurre?

$$\begin{aligned} &\exists x (\text{EsMoro}(x) \wedge \forall y (\text{EsCristiano}(y) \rightarrow \text{MasFuerte}(x,y))) \\ &\exists x \forall y (\text{EsMoro}(x) \wedge (\text{EsCristiano}(y) \rightarrow \text{MasFuerte}(x,y))) \\ &\exists x \forall y ((\text{EsMoro}(x) \wedge \text{EsCristiano}(y)) \rightarrow \text{MasFuerte}(x,y)) \end{aligned}$$

La Práctica

Además de para la evaluación sintáctica y semántica de fórmulas lógicas, este programa nos puede ayudar a entender y practicar con otros conceptos lógicos: equivalencia de fórmulas, fórmulas complementarias, concepto semántico de deducción correcta, método del contraejemplo, ... En este apartado vamos a realizar algunos ejercicios que nos permitan repasar algunas de estas nociones.

1. Equivalencia de fórmulas

a) Escribe las *fórmulas bien formadas* correspondientes a las siguientes sentencias y las guardas en el fichero “practica4a.sen”. Evalúalas sintácticamente para comprobar que realmente están bien escritas.

- Si *b* es cristiano entonces lleva espada
- Solo si *b* es cristiano llevará espada

- $O b$ no es cristiano o lleva espada
- $O b$ es cristiano o no lleva espada
- b es cristiano con espada
- No es cierto que b sea un cristiano que no lleve espada

Abre un mundo nuevo ("practica4a.mun") e inserta los individuos que creas necesarios. Comprueba qué sentencias son equivalentes. Haz modificaciones en el mundo para ver el efecto sobre el valor de verdad de dichas sentencias.

b) Escribe las *fórmulas bien formadas* correspondientes a las siguientes sentencias y las guardas en el fichero "practica4b.sen". Evalúalas sintácticamente para comprobar que realmente están bien escritas.

- No es cierto que a sea un moro que monte a caballo
- Ni a es un moro ni monta a caballo
- a es un moro que monta a caballo
- $O a$ no es moro o no monta a caballo
- No es cierto que a sea un moro o que monte a caballo
- $O a$ es moro o monta a caballo

Abre un mundo nuevo ("practica4b.mun") e inserta los individuos que creas necesarios. Comprueba qué sentencias son equivalentes. Haz modificaciones en el mundo para ver el efecto sobre el valor de verdad de dichas sentencias.

c) Escribe las *fórmulas bien formadas* correspondientes a las siguientes sentencias y las guardas en el fichero "practica4c.sen". Evalúalas sintácticamente para comprobar que realmente están bien escritas.

- Todos los individuos son no moros
- No existe ningún individuo que sea moro
- Algún individuo no es moro
- No todos los individuos son moros
- No existe ningún individuo que no sea moro
- No todos los individuos son no moros

Abre un mundo nuevo ("practica4c.mun") e inserta los individuos que creas necesarios. Comprueba qué sentencias son equivalentes. Haz modificaciones en el mundo para ver el efecto sobre el valor de verdad de dichas sentencias.

La Tabla de las Figuras

Ilustración 1: Ventana "Acerca de ..."	2
Ilustración 2: Pantalla principal del programa	3
Ilustración 5: Ventana de mundo: Tablero	7
Ilustración 6: Tipos de individuos	7
Ilustración 7: Insertar una nueva pieza	8
Ilustración 8: Tablero con pieza	8
Ilustración 9: Insertar pieza desde tablero	8
Ilustración 10: Eliminar pieza	9

La Bibliografía

- [Barwise92] JON BARWISE y JOHN ETCEHEMENDY. *The Language of First-Order Logic*. CSLI Lecture Notes number 34, Center for the Study of Language and Information, Stanford, 1992.
- [Barwise98] JON BARWISE y JOHN ETCEHEMENDY. *Computers, visualization, and the nature of reasoning*. Center for the Study of Language and Information, Stanford, 1998.
- [Goldson93] DOUG GOLDSON, STEVE REEVES y RICHARD BORNAT. *A Review of Several Programs for the Teaching of Logic*. The Computer Journal, vol. 36, nº 4, 1993.
- [Llorens96] FARAÓN LLORENS LARGO y M^a JESÚS CASTEL DE HARO. *Lógica de Primer Orden*. Departamento de Tecnología Informática y Computación, Universidad de Alicante, 1996.
- [Llorens98] FARAÓN LLORENS LARGO y M^a JESÚS CASTEL DE HARO. *Lògica de Primer Ordre*. Col·lecció Joan Fuster 5, Materials per a la docència en valencià, Secretariat de Normalització Lingüística, Universitat d' Alacant, 1998.